Windows Programming

Visual C++ MFC Programming

Lecture 05

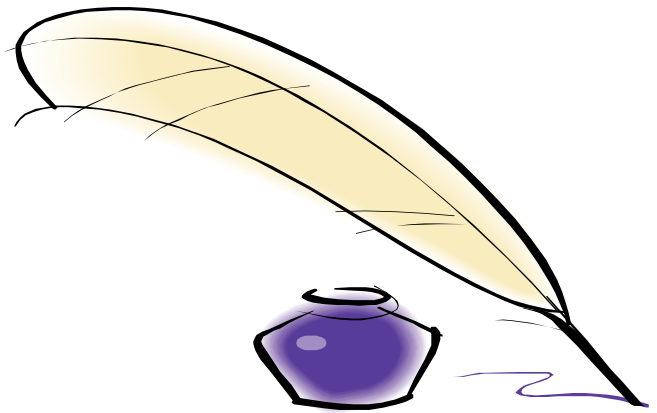
김예진
Dept. of Game Software

Notices

- 03/07: 502 → 501 등록 이동
- 03/21: HW 1 (Due: 03/28)

Plan

- MFC 화면 출력
  - GDI Object
  - CPen
  - CBrush
  - CFont
  - CRgn
  - CBitmap
  - Double Buffering



GDI Object

- 기본값:

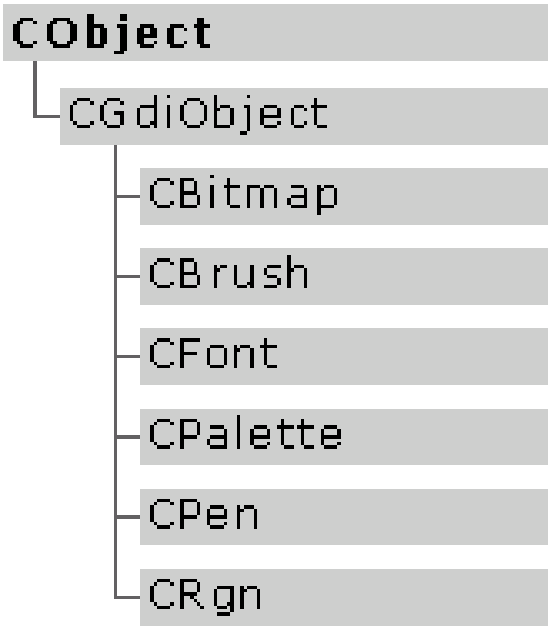
- 색을 바꾸거나 특성을 바꿀 수 없을까?
  - Pen의 특성을 바꾼다 : 선
  - Brush의 특성을 바꾼다: 면

GDI Object

- GDI object(객체)
  - GDI에서 출력할 때 사용하는 도구

- 종류

GDI Object	용도	Class
Pen	선을 그릴 때	CPen
Brush	면의 내부를 채울 때	CBrush
Font	문자를 출력할 때	CFont
Bitmap	Pixel의 집합으로 이루어진 그림을 다룰 때	CBitmap
Palete	출력될 색의 집합을 다룰 때	CPalette
Region	다양한 형태의 면을 정의할 때	CRgn



GDI Object

- 그림을 그리는 순서 (일반적인 경우)
  1. Pen을 고름
  2. Pen을 쥠
  3. 그림을 그림
  4. Pen을 내려놓음

- 그림을 그리는 순서 (GDI객체사용)
  1. 사용할 객체(pen)를 정의
  2. DC에 이 객체를 지정: CDC::SelectObject()
     - 이전에 가지고 있던 객체를 임시저장
  3. 그림을 그림
  4. 사용할 객체를 선택해제
     - 이전에 가지고 있던 객체로 환원

CPen

- Pen 생성 방법

style width color

```
// 방법 1
CPen pen(PS_SOLID, 2, RGB(255, 0, 0));                    // 생성자

// 방법2
CPen pen;
pen.CreatePen(PS_SOLID, 2, RGB(255, 0, 0));    // 초기화함수
```

- Pen style

PS_SOLID

PS_DASH

PS_DOT

PS_DASHDOT

PS_DASHDOTDOT

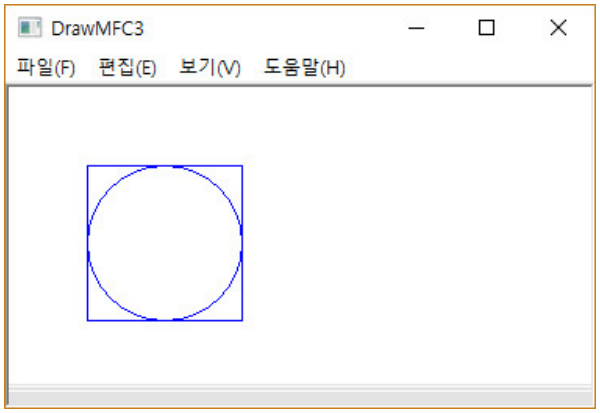
PS_NULL

PS_INSIDEFRAME

CPen

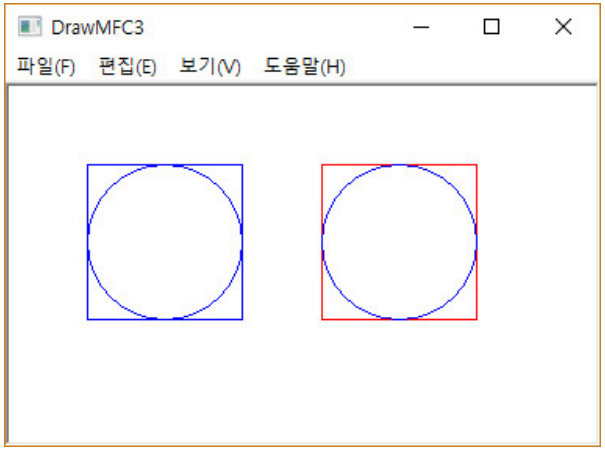
- 사용 예: Blue pen 사용

```
CPen pen1(PS_SOLID, 1, RGB(0, 0, 255));
dc.SelectObject(&pen1);
dc.Rectangle(50, 50, 150, 150);
dc.Ellipse(50, 50, 150, 150);
```



- 사용 예: Blue pen 저장 후 red pen 사용

```
CPen pen2(PS_SOLID, 1, RGB(255, 0, 0));
CPen *pOldPen = dc.SelectObject(&pen2);
dc.Rectangle(200, 50, 300, 150);
dc.SelectObject(pOldPen);
dc.Ellipse(200, 50, 300, 150);
```



CBrush

- Brush
  - Face(면)을 어떻게 채우는가를 정의하는 것

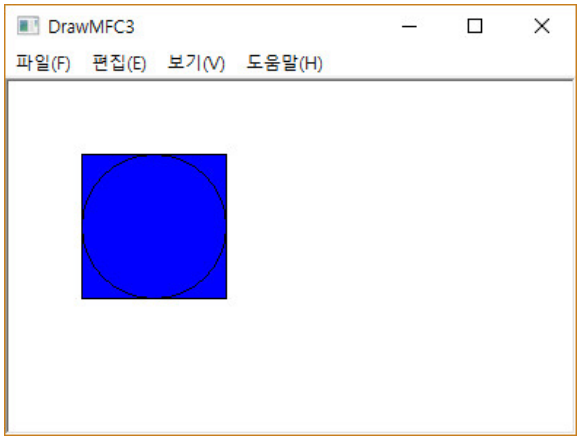
- Brush type: 생성자에 따라 결정됨

Brush type	생성 예
솔리드(Solid, 속이 채워짐)	CBrush brush(RGB(255, 0, 0));
해치(Hatch, 교차된 평행선 무늬)	CBrush brush(HS_DIAGCROSS, RGB(255, 0, 0));
패턴(Pattern, 비트맵의 반복 무늬)	CBitmap bitmap; bitmap.LoadBitmap(IDB_BITMAP1); CBrush brush(&bitmap);

CBrush

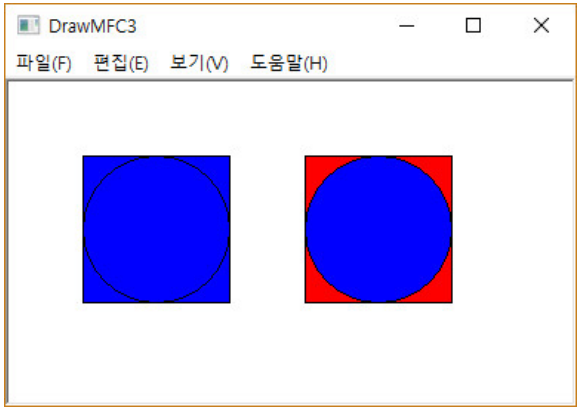
- 사용 예: Blue brush 사용

```
CBrush brush1(RGB(0, 0, 255));
dc.SelectObject(&brush1);
dc.Rectangle(50, 50, 150, 150);
dc.Ellipse(50, 50, 150, 150);
```



- 사용 예: Blue brush 저장 후 red brush 사용

```
CBrush brush2(RGB(255, 0, 0));
CBrush *pOldBrush = dc.SelectObject(&brush2);
dc.Rectangle(200, 50, 300, 150);
dc.SelectObject(pOldBrush);
dc.Ellipse(200, 50, 300, 150);
```



CFont

- 생성 방법
  - CFont 객체 생성
  - CFont 객체에 대해 CreateFont() 함수를 호출

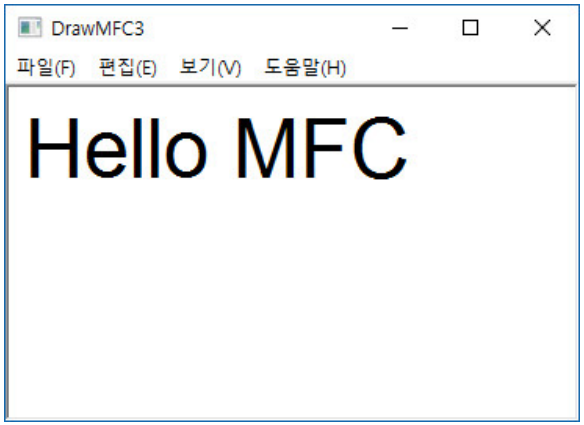
```
CFont font;
font.CreateFont(...);
// font.CreateFontIndirect(...);
// font.CreatePointFont(...);
// font.CreatePointFontIndirect(...);
```

CFont

- 사용 예: CFont 사용

크기 글씨체이름

```
CFont font;
font.CreatePointFont(400, _T("Arial"));
dc.SelectObject(&font);
dc.TextOut(10, 10, _T("Hello MFC"));
```

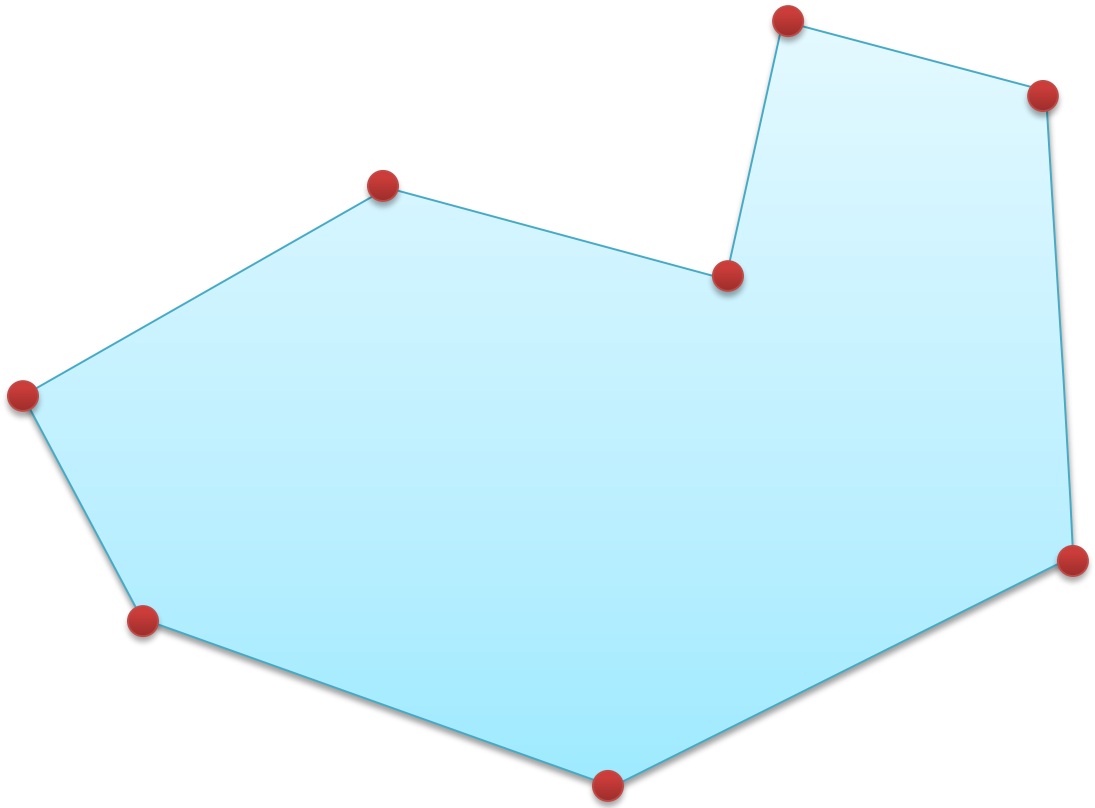


- 미리 정의된 내장 객체

이름	용도
BLACK_PEN	폭이 1 pixel인 검정색 pen
WHITE_PEN	폭이 1 pixel인 흰색 pen
NULL_PEN	투명 pen
BLACK_BRUSH	검정색 brush
GRAY_BRUSH	회색 brush
HOLLOW_BRUSH NULL_BRUSH	투명 brush
SYSTEM_FONT	윈도우 운영체제가 사용하는 font 예) menu, 대화상자, ...

CRgn

- Polygon(다각형)



점들의 집합으로 정의됨

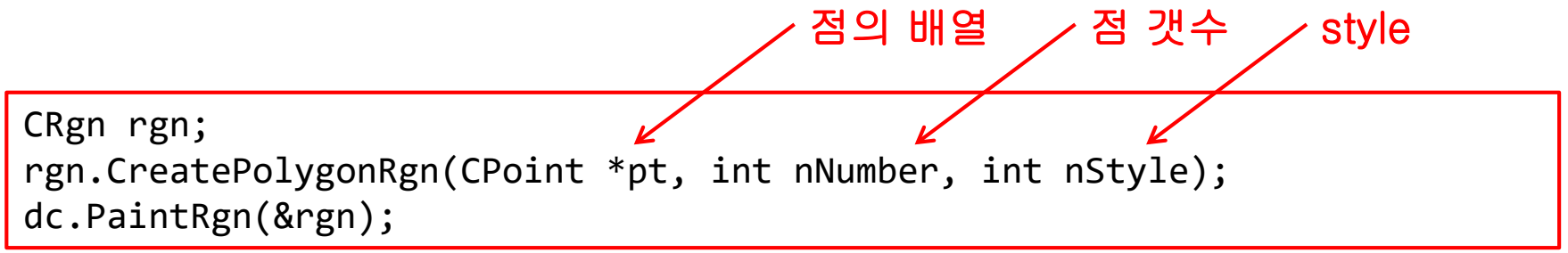
CRgn

- CPoint 사용 방법
  - 2차원 위치 정보를 표현하는 class
  - Member 변수: x, y

```
// 위치 정보를 멤버변수에 할당
CPoint pt1;
pt1.x = 100;
Pt1.y = 200;

// 생성자를 통한 할당
CPoint pt2(300, 200);

// 할당연산자를 통한 할당
CPoint pt3;
pt3 = pt2;

// Copy 생성자를 통한 할당
CPoint pt4(pt1);
```

CRgn

- CRgn 사용 방법
  - nStyle: ALTERNATE or WINDING

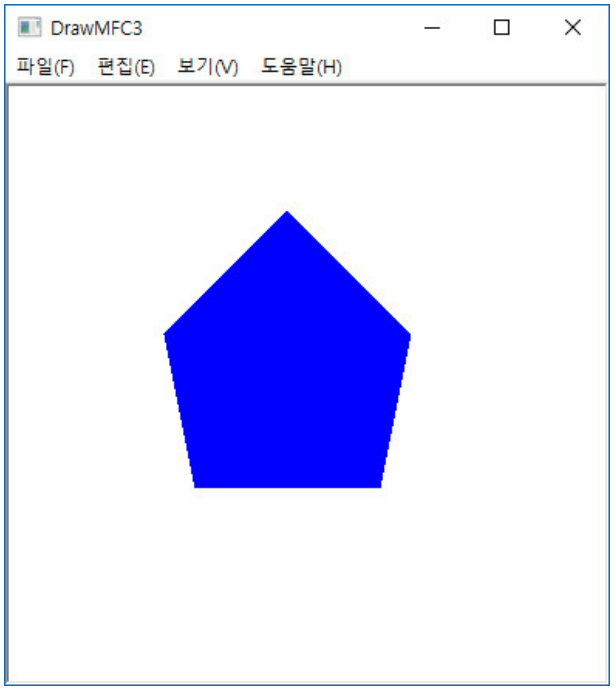



```
CRgn rgn;
rgn.CreatePolygonRgn(CPoint *pt, int nNumber, int nStyle);
dc.PaintRgn(&rgn);
```

CRgn

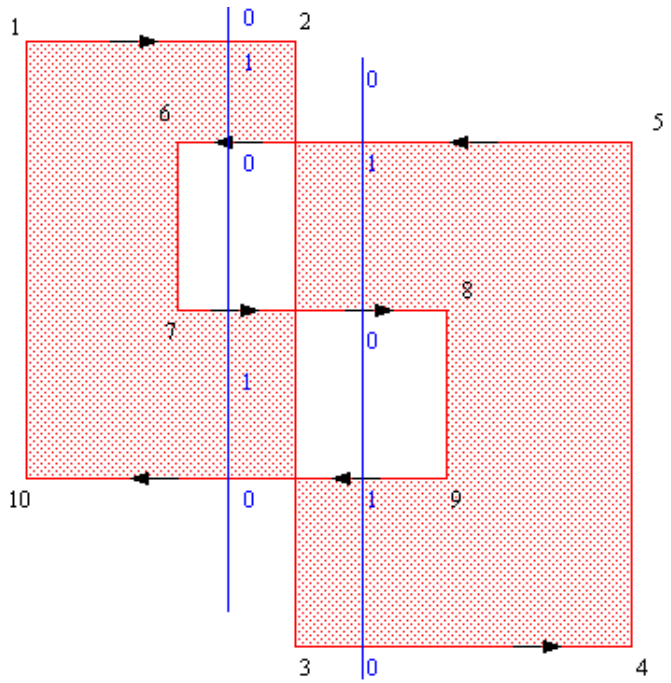
- 사용 예: Blue 색상의 pentagon(오각형) 그리기

```
CRgn   rgn;
CPoint ptVertex[5];
ptVertex[0] = CPoint(180,80);
ptVertex[1] = CPoint(100,160);
ptVertex[2] = CPoint(120,260);
ptVertex[3] = CPoint(240,260);
ptVertex[4] = CPoint(260,160);

rgn.CreatePolygonRgn( ptVertex, 5, ALTERNATE);
CBrush brush(RGB(0,0,255));
dc.SelectObject(&brush);
dc.PaintRgn(&rgn);

// or

dc.FillRgn(&rgn, &brush);
```

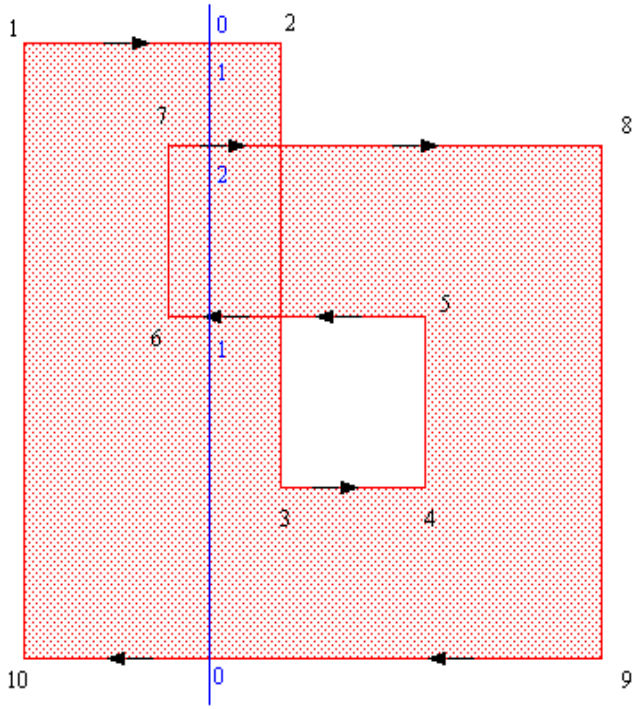


CRgn

- ALTERNATE or WINDING
  - 면의 내부를 채우는 알고리즘의 선택



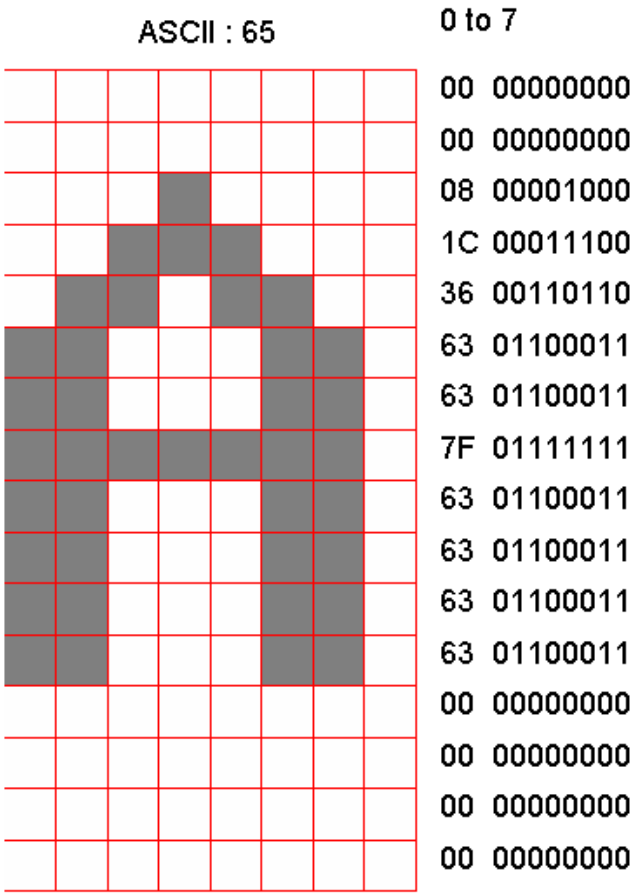
Alternate filling

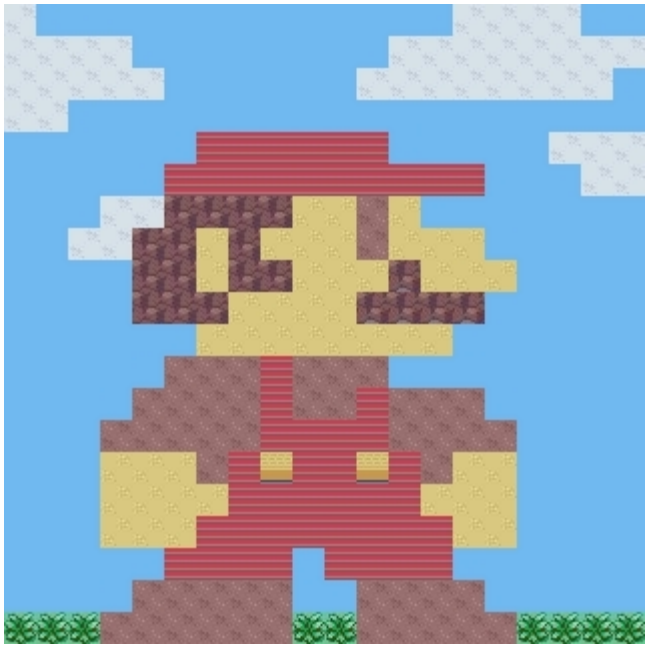


Winding filling

CBitmap

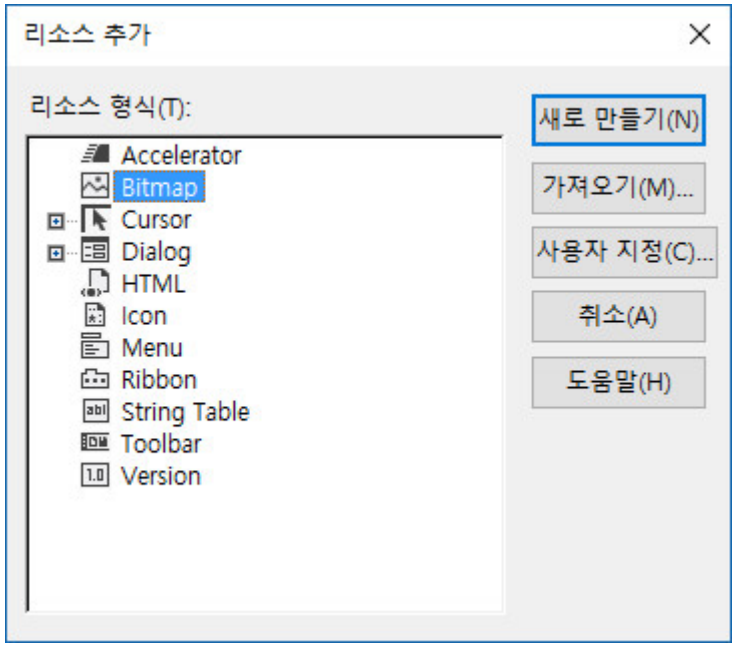
- Bitmap
  - 그림을 dot(pixel)로 표현 하는 것

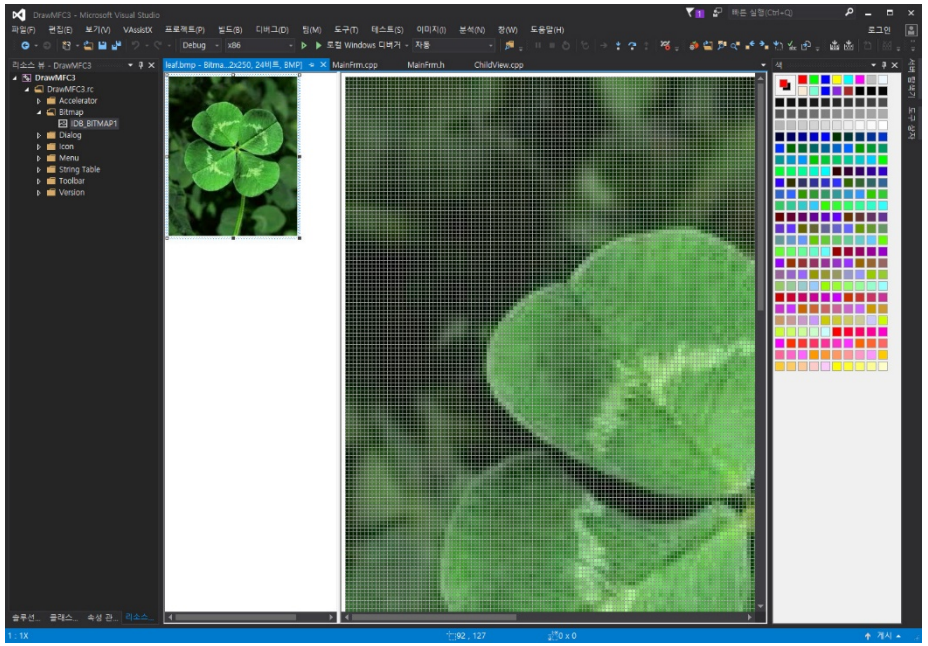




CBitmap

- 사용 예: 간단한 bitmap 만들기
  - 리소스 뷰에서 리소스 추가를 사용





CBitmap

- 사용 예: Brush에 로딩

리소스 아이디

```
CBitmap bitmap;
bitmap.LoadBitmap(IDB_BITMAP1);

CBrush brush(&bitmap);
dc.SelectObject(&brush);
dc.Rectangle(0, 0, 200, 200);
```

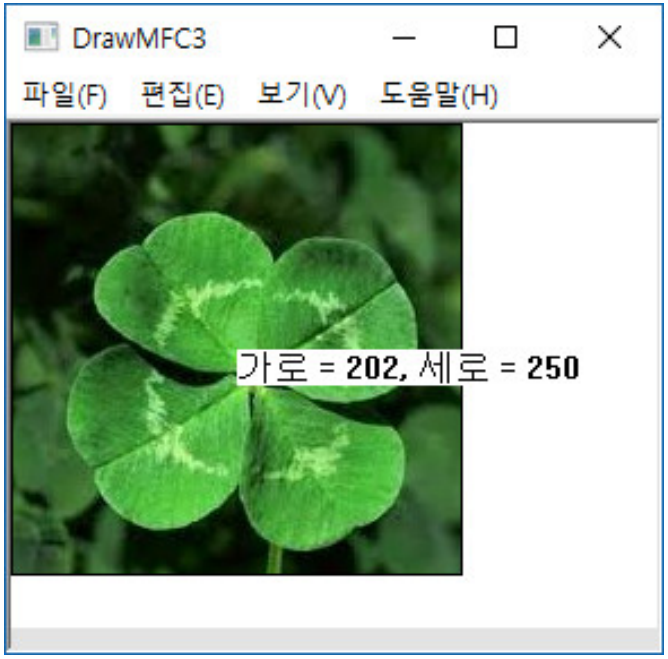
- Bitmap 정보

```
int CBitmap::GetBitmap(BITMAP* pBitMap) ;

struct BITMAP {
    int bmType;
    int bmWidth;            // 비트맵의 폭(픽셀 단위)
    int bmHeight;           // 비트맵의 높이(픽셀 단위)
    int bmWidthBytes;
    BYTE bmPlanes;
    BYTE bmBitsPixel;
    LPVOID bmBits;
};
```

CBitmap

- 사용 예: Bitmap 정보 출력

```
// 이어서
CBitmap bitmap;
bitmap.LoadBitmap(IDB_BITMAP1);
BITMAP bmpinfo;
bitmap.GetBitmap(&bmpinfo);

// CString: VC++에서 지원하는 문자열 클래스
CString str;
// Format(..): printf와 같은 기능을 하는 맴버 함수
str.Format(_T("가로 = %d, 세로 = %d\n"),
     bmpinfo.bmWidth, bmpinfo.bmHeight);

dc.TextOut(100, 100, str);
```



CBitmap

- Bitmap 직접 출력
  - 한 점씩 그리지 않고 한꺼번(**block**)에 메모리로 보냄(**transfer**)

- Memory에 그림 그리고 한꺼번에 넘겨주기 절차
  1. 원래의 도화지(dc) 준비 — **CPaintDC dc;**
  2. 다른 도화지(메모리 DC) 준비 — **CDC::CreateCompatibleDC( );**
  3. 다른 도화지에 그림 그리기 — **CDC::SelectObject( );**
  4. 원래의 도화지로 그린거 오려 붙이기 — **dc.BitBlt( );**

  *BitBlt: Bit Block Transfer

Bitmap

- 사용 예: 지정 위치에 bitmap의 원래 크기로 오려 붙이기
  - 위치: (100, 100)

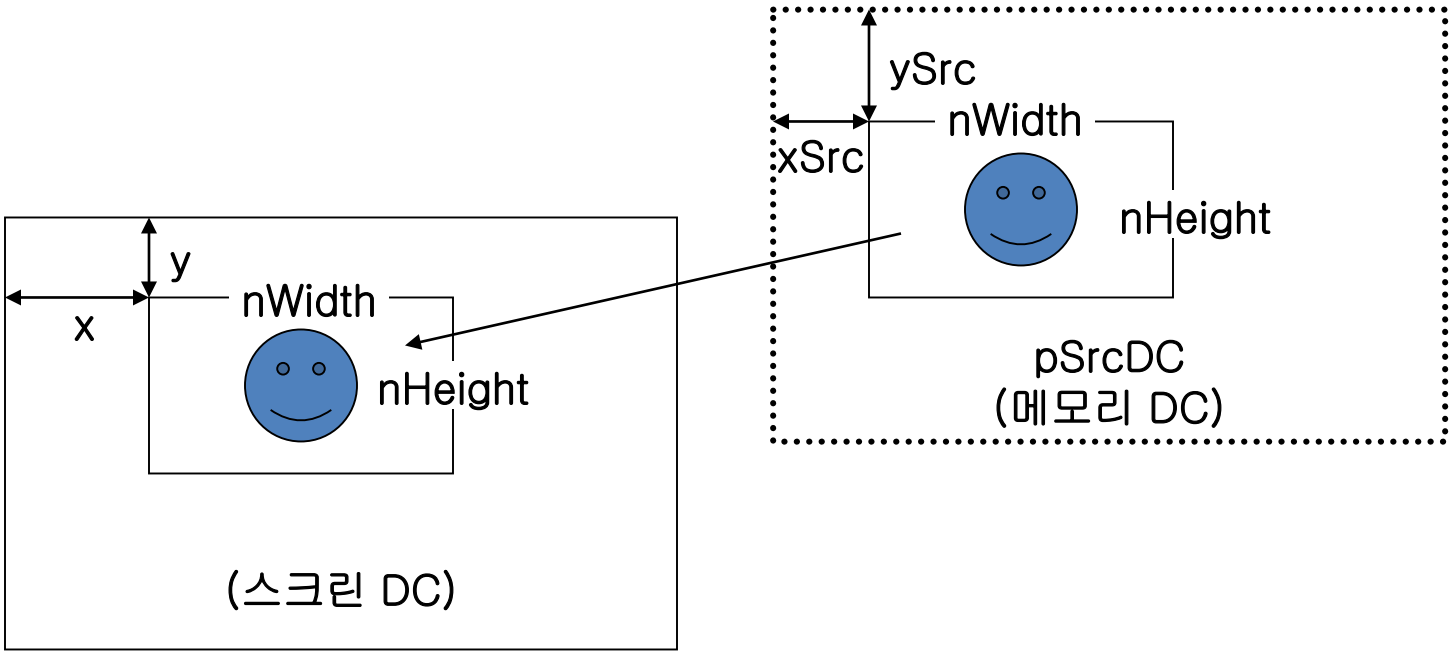
```
CPaintDC dc(this);

CBitmap bitmap;
bitmap.LoadBitmap(IDB_BITMAP1);
BITMAP bmpInfo;
bitmap.GetBitmap(&bmpInfo);

CDC memDc;
memDc.CreateCompatibleDC(&dc);
memDc.SelectObject(&bitmap);

dc.BitBlt(100, 100,
     bmpInfo.bmWidth, bmpInfo.bmHeight,
     &memDc, 0, 0, SRCCOPY);
```

CBitmap

- BitBlt 출력 함수

```
BOOL BitBlt (int x, int y, int nWidth, int nHeight, CDC* pSrcDC, int xSrc, int ySrc,
          DWORD dwRop);
```

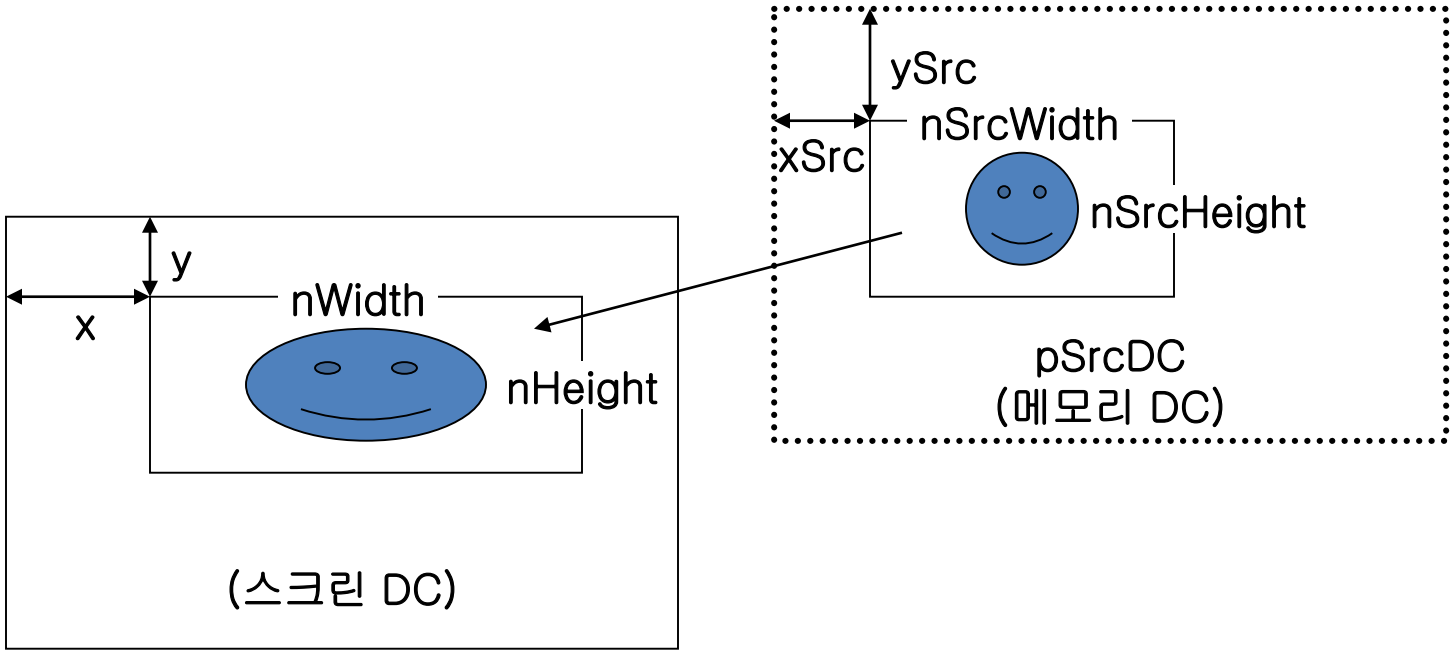
그림을 복사하는 방법: SRCCOPY



CBitmap

- StretchBlt 출력 함수

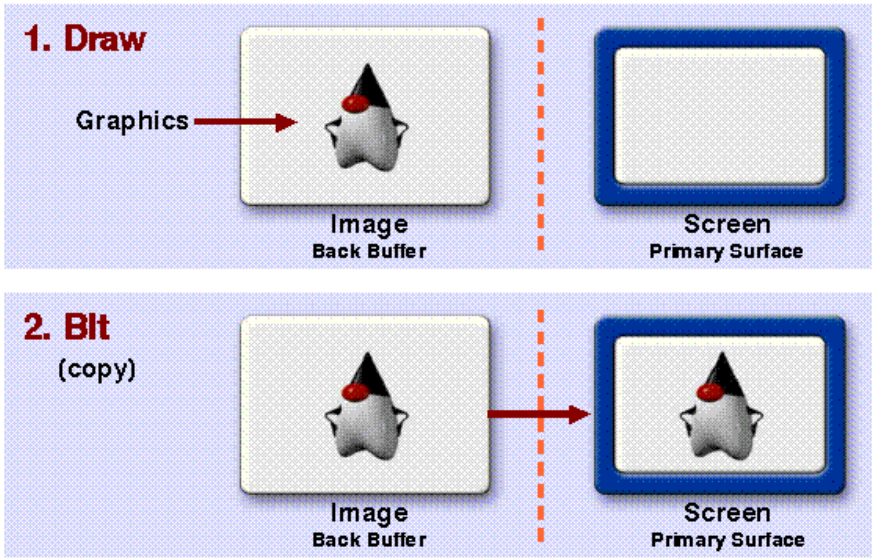
```
BOOL StretchBlt (int x, int y, int nWidth, int nHeight, CDC* pSrcDC, int xSrc, int
        ySrc, int nSrcWidth, int nSrcHeight, DWORD dwRop) ;
```



Double Buffering

- 같은 memory로 그림을 그리고 출력하는 방식의 문제점
  - 그림을 다시 그릴 때 그리는 과정이 보임
    - 특히, 느린 CPU를 사용할 때
  - 그림을 다시 그릴 때 깜박임
    - 원래 내용을 invalidate(무효화): 먼저 하얗게 지우고 다시 그림

- 두 개의 Device Context(도화지)를 이용
  1. 그림 그리는 용도
     - Back buffe에 그림을 그림
  2. 그림 출력 용도
     - Front(Screen) buffer로 출력



Double Buffering

- Memory에 그림 그리고 한꺼번에 넘겨주기 절차
  1. Memory DC 만들기: CreateCompatibleDC( )
  2. 그림 그릴 memory 만들기: CreateCompatibleBitmap( )
  3. DC와 bitmap을 연결 SelectObject( )
  4. 그림 그리기
  5. 그려진 그림을 화면 DC로 전달: BitBlt( )

  ➔ Double buffering: 그림을 다른 도화지에 그리고 도화지를 빠르게 바꿔 치는 기법

Double Buffering

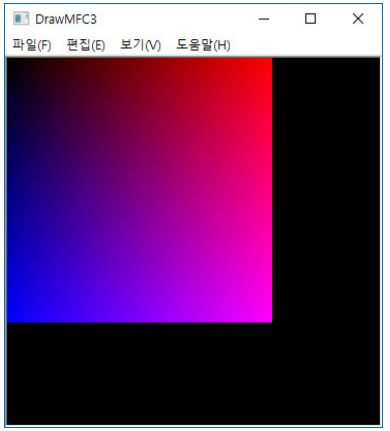
- 사용 예: Color가 점점 변하는 사각형을 출력하기 (1/3)
  - Color: Blue와 red 사이
  - 빠른 출력을 위해 double buffering을 사용

```
for(int x = 0 ; x < 256 ; x++)
    for(int y = 0 ; y < 256 ; y++)
        dc.SetPixelV(x, y, RGB(x, 0, y));
```

Double Buffering

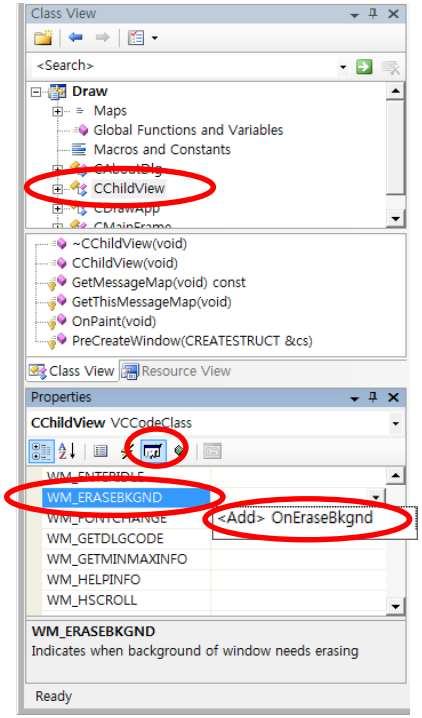
- 사용 예: Color가 점점 변하는 사각형을 출력하기 (2/3)
  - 빠른 출력을 위해 double buffering 사용 예

```
CRect rect;
GetClientRect(rect);

CDC memDC;          // 가상 DC
memDC.CreateCompatibleDC(&dc);

CBitmap bitmap;     // 그림을 저장할 공간 마련
bitmap.CreateCompatibleBitmap(&dc, rect.Width(), rect.Height());
memDC.SelectObject(&bitmap);

for(int x = 0 ; x < 256 ; x++)
     for(int y = 0 ; y < 256 ; y++)
          memDC.SetPixelV(x, y, RGB(x, 0, y));

dc.BitBlt(0, 0, rect.Width(), rect.Height(), &memDC, 0, 0, SRCCOPY);
```



Double Buffering

- 사용 예: Color가 점점 변하는 사각형을 출력하기 (3/3)
  - Double buffering 사용시 그림이 빨리 그려짐
    - 그러나, 여전히 깜박임 → 화면을 매번 깨끗이 지우기 때문!
  - "화면을 지울 필요가 있다"라는 message를 받아서 지우지 않음
    - **WM_ERASEBKGND:** 화면을 지울 때(하얗게 칠할 때) 생성
    - WM_ERASEBKGND message 처리기를 생성 후 아무 일도 안하게 수정




```
BOOL CChildView::OnEraseBkgnd(CDC* pDC)
{
        //  return CWnd::OnEraseBkgnd(pDC);
        return true;
}
```

Q & A